Indian Journal of Emerging Electronics in Computer Communications, Vol.1, Issue 2, Dec 2014, ISSN: 2393-8366

Skyline Query Processing Using Localized Servers

M. V.Ezhil Dyana[1], B. Nithlesh[2], K. Raguvaran[2]

[#1]Assistant Professor, Dept of CSE, Alpha College of Engg, Chennai, T.N, India.
*[*2]U.G Students, B.E CSE, Alpha College of Engg, Chennai, T.N, India.*

[1]dyanaezhil@gmail.com, [2]nithlesh123@gmail.com, [2]raguvaran2404@gmail.com

Abstract - The literature on skyline algorithms has so far dealt mainly with queries of static query points over static data sets. With the increasing number of mobile service applications and users, however, the need for continuous skyline query processing has become more pressing. A continuous skyline query involves not only static dimensions, but also the dynamic one. In this paper, we examine the spatiotemporal coherence of the problem and propose a continuous skyline query processing strategy for moving query points. First, we distinguish the data points that are permanently in the skyline and use them to derive a search bound. Second, we investigate the connection between the spatial positions of data points and their dominance relationship, which provides an indication of where to find changes in the skyline and how to maintain the skyline continuously.

Keywords – I-SKY, N-SKY, LBS, SSQ.

I. INTRODUCTION

Location-based services (LBS) are a general class of computer program-level services used to include specific controls for location and time data as control features in computer programs. As such LBS is an information service and has a number of uses in social networking today as an entertainment service, which is accessible with mobile devices through the mobile network and which uses information on the geographical position of the mobile device. This has become more and more important with the expansion of the smartphone and tablet markets as well LBS include services to identify a location of a person or object, such as discovering the nearest banking cash machine (a.k.a. ATM) or the whereabouts of a friend or employee. LBS include parcel tracking and vehicle tracking services. LBS can include mobile commerce when taking the form of coupons or advertising directed at customers based on their current location. They include personalized weather services and even location-based games Conventional LBSs focus on processing proximity-based queries, including the range query and nearest neighbor (NN) query. However, these queries are not sufficient for the applications that need to consider both spatial and non-spatial attributes of the objects being queried.

A typical scenario is finding a nearby car park with cheap parking fee, in which distance is a spatial attribute and parking fee is nonspatial attribute. Clearly, here a multicriterion query is more appealing than a conventional spatial query that considers the distance only. Among various multicriterion queries, the skyline query is considered as one of the most classical ones and receives the most attention in LBS research. However, the dynamic nature of the spatial attribute makes skyline queries in LBSs unique and challenging. Take the above park-finding scenario, for example. At different locations, the distances from the user to the car parks are not the same. As a consequence, the skyline query results differ for different locations. To efficiently compute location-based skyline results, a number of algorithms have been proposed for one-shot queries, and continuous queries. Nevertheless, these previous studies have limitations as they simply assume that the query location is an exact location point or a line segment.

We relax this assumption and propose a more general skyline query—range-based skyline query (RSQ), which takes a spatial range as the input of user location, as opposed to a point or a line in existing LBS skyline studies. Compared to existing skyline queries, the range-based skyline query might be more practical for several reasons:

- The location of the query issuer could be shaped as a spatial region (e.g., a region that covers a complex spatial object or a group of users).
- Due to limited precision of localization devices, the query issuer does not have accurate knowledge about his/her exact location.

- For privacy reasons, the query issuer may not want to expose his/her exact location to the service provider. And a widely used solution is to blur the location into a cloaking region so that the adversary has no clue about where the query issuer is exactly located.

II. RELATED WORKS

In essence, a range-based skyline query inherits the characteristics of a skyline query and a range-based query. As such, we review the existing work on these two queries.

Skyline query processing. Skyline query processing was first introduced into the database community by Borzonyi et al. A number of algorithms have been proposed from then on. These algorithms can be divided into two categories. The first category is nonindex algorithms and the representatives are Black Nested Loop (BNL) and Divide-and-Conquer (D&C). BNL scans the data set sequentially and compares each new object to all skyline candidates kept in the memory. D&C partitions the data set into several parts, processes each part in the memory, and, finally, merges all partial skylines together. SFS improved BNL by presorting the data sets. In the Bitmap approach, each data point is encoded in a bit string and the skyline is computed by some efficient operations on the bit matrix of all data points.

The other category of skyline algorithms is index-based. A high-dimensional data set is converted into a 1D data set and a B+-tree is built to accelerate query processing. An algorithm called NN was proposed based on the depthfirst nearest neighbor search via R_-tree. Papadias et al proposed an improved algorithm, called Branchand-Bound Skyline (BBS), which was based on the best-first nearest neighbor search. By accessing only the nodes that contain skyline points and employing effective pruning techniques, BBS achieves the optimal I/O access. More recently, a subset of skyline points are collected to approximately represent the distribution of an entire set of skyline points. Lee et al proposed a new index structure called ZBtree to index and store data points based on Z-order curve, and developed a novel algorithm ZSearch to process skyline queries. Huang et al introduced the skyline query problem in the context of LBSs and proposed a continuous skyline query processing algorithm called CSQ for moving clients. Assuming a linear movement model, CSQ processes the skyline query at the starting point of the query segment and tries to predict the possible changes to the answer set when the client moves.

This avoids continuously computing the skyline results from scratch. Zheng et al introduced a notion of valid scope for LBS skyline queries, which saves the recomputation if the next query point is still inside the valid scope. Sharifzadeh and Shahabi defined a variant of skyline query in LBSs by considering the distance between an object and a set of query points. Our work is inspired by these prior point-based or line-based skyline algorithms, but focus on range-based skyline queries. Obviously, range-based skyline queries cannot be processed by simply applying the existing algorithms because the number of query points/line segments in a range is infinite.

Range-based query processing. Range-based query processing has recently received notable attention as the location privacy issue is becoming increasingly important. For privacy reasons, mobile clients tend to blur their exact locations into an uncertain range so that the service provider cannot find where they are exactly located. The service provider then returns a superset of candidate results for every possible query point in the range. Finally, the clients filter these results and obtain the true result by their exact locations. Existing range-based query algorithms studied only range-based kNN (RkNN) query. Hu and Lee proposed the first RkNN solution for rectangular ranges. Ku et al studied the same problem in spatial networks. Complementally, Xu et al developed an RkNN algorithm for circular ranges. To the best of our knowledge, there is no work that has studied range-based skyline queries.

III. DESIGN AND IMPLEMENTATION

A. Outline and Architecture of the skyline queries

In this project, we propose a method how to process the location based client request resources by using the support of algorithms. There are two types of algorithms

- a. (I-sky)Index based skyline query processing.
- b. (N-sky)Not index based skyline query processing.

By using the index based algorithm we may calculate the pre-index of the object motion and we may give the results for the particular query (or) resource. And by using this index query processing is high. But, by using the index based algorithm we may get some expensive. So because of that reason we may go to the N-sky index based skyline query processing which is not having any index.

Thus, given a range-based skyline query, the service provider should return a collective set of skyline results for every possible query point of the user in R. Unfortunately; this is not easy as the number of possible points in R is infinite. To address this issue, we propose a novel index-based algorithm, called I-SKY. The idea is to precompute the skyline scopes for all objects by their dominance relations. By indexing the skyline scopes, a range-based skyline query can be efficiently processed through a root-to-leaf traversal of the index tree. The range-based skyline problem becomes even more challenging for dynamic data sets where the objects being queried can move and update their spatial attributes frequently. In this function first a sender can send the data what he wants to the receiver through the number of localized servers and when the receiver is moving always in an environment then also we can send the data by using the number of localized servers and it will calculate distance of the number of localized servers from the source to the destination and chooses the best one among which is available based on the requirement and finally sent to the receiver.

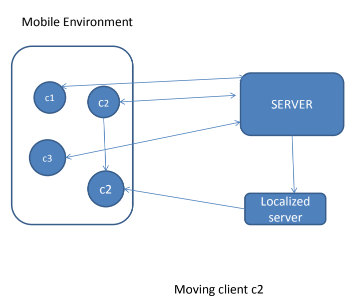
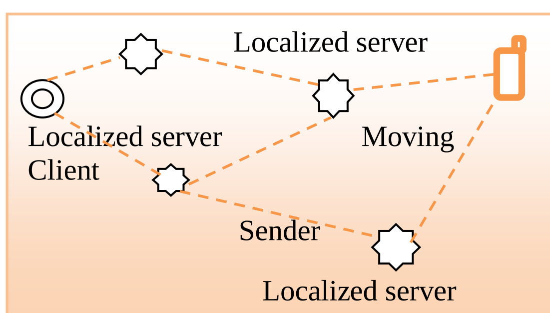


Fig .1. System Architecture



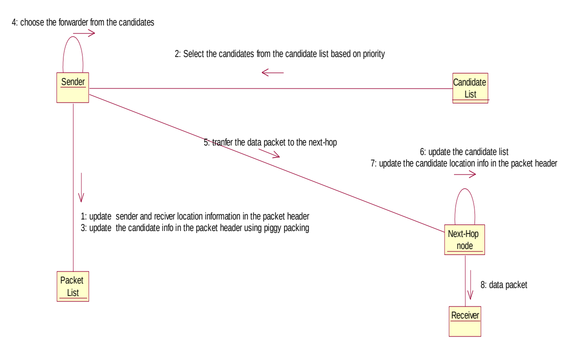


Fig. 2.collaboration diagram

IV. CONCLUSION & FUTURE WORK

The experimental results show that our proposed algorithms outperform than the existing line-based skyline solution in terms of both the CPU time and I/O cost. As for future work, we will extend the query range to arbitrary shapes that have a closed-form mathematical expression, especially those with arc-like boundaries. Furthermore, we plan to extend our range-based skyline problem to road networks.

REFERENCES

[1] J.L. Bentley, H.T. Kung, M. Schkolnick, and C.D. Thompson, "On the Average Number of Maxima in a Set of Vectors and Applications," J. ACM, vol. 25, no. 4, pp. 536-543, 1978.

[2] M.Berg, O.Cheong, and M.Kreveld, Computational Geometry:Algorithms and Applications, third ed., chapter 7. Springer, 2008.

[3] S. Borzonyi, D. Kossmann, and K. Stocker, "The Skyline Operator," Proc. Int'l Conf. Data Eng., pp. 421-430, 2001.

[4] Y. Cai and T. Xu, "Design, Analysis, and Implementation of a Large-Scale Real-Time Location-Based Information Sharing System," Proc. ACM Sixth Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '08), pp. 106-117, 2008.

[5] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with Pre-Sorting," Proc. Int'l Conf. Data Eng., 2003.

[6] C.-Y. Chow, M.F. Mokbel, and W.G. Aref, "Casper*: Query Processing for Location Services without Compromising Privacy," ACM Trans. Database Systems, vol. 34, no. 4, article 24, Dec. 2009.

[7] B. Fan and H. Ahmed, "A Survey of Mobility Models in Wireless Ad-Hoc Networks," Wireless Ad-Hoc Networks, Chapter 1, Kluwer Academic, 2006.

[8] Y. Gao and B. Zheng, "Continuous Obstructed Nearest Neighbor Queries in Spatial Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 557-590, 2009.

[9] B. Gedik, K.-L. L, P.S. Yu, and L. Liu, "Processing Moving Queries over Moving Objects Using Motion-Adaptive Indexes ," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 5, pp. 651-668, May 2006.

[10] H. Hu and D.L. Lee, "Range Nearest-Neighbor Query," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 1, pp. 78-91, Jan. 2006.